An Improved TrICP Point Cloud Registration Method Based on Automatically Trimming Overlap Regions

Pengcheng Jiang and Yuan Li(✉)

Beijing Institute of Technology, Beijing 100081, China
liyuan@bit.edu.cn

Abstract. To address the challenge of determining the parameters of registration algorithms under low point clouds overlap, which hinders automatic and efficient calculations, we introduce an improved variant of the TrICP algorithm capable of automatically extracting the overlap regions. Firstly, the triangle threshold method is used to estimate the distance threshold, and the overlap region is extracted and bidirectionally merged to obtain relatively complete overlap point clouds. To mitigate the risk of getting stuck in local optimal solutions and minimize the impact of incorrectly identified point pairs in non-overlap regions, we incorporate an effectiveness factor in the calculation of Singular Value Decomposition (SVD) to weight the importance of the point pairs. To decrease the overlap point clouds extraction times and reduce the associated time costs, we implemented multiple iterations following each extraction of the overlap point clouds. We compared our algorithm with the ICP and TrICP algorithms using publicly available point clouds data and demonstrated the effectiveness of our algorithm in automatically addressing the challenge of fine registration for point clouds with low overlap.

Keywords: Point Cloud Registration · Improved TrICP · Triangular Threshold Method · Effectiveness Factor

1 Introduction

3D imaging can be obtained by means such as stereoscopic cameras [1] and cameras based on motion recovery structures [2], which have a wide range of applications in 3D reconstruction, computer vision and other fields. Due to the limitation of equipment, one imaging may not be able to completely collect the data. In this case, it is particularly important to register and splice point clouds at different viewing angle. Point cloud registration technology can be applied in many fields, including map construction and patient image synthesis in medical field [3], etc., which need point cloud registration algorithms that offer both high accuracy and efficiency. Therefore, it is of great meaning to study automatic, quick and accurate registration algorithm. The key points of this paper is on addressing the challenge of achieving fine registration for partially overlap point clouds.

B. Xin et al. (Eds.): IWACIII 2023, CCIS 1932, pp. 70–80, 2024.
https://doi.org/10.1007/978-981-99-7593-8_7

In the problem of registering point clouds, Besl et al. [4] proposed ICP(Iterative Closest Point) algorithm in a pioneering way, which has excellent effect on point clouds with high overlap and good initial location [5]. However, point clouds with different viewing angles are partially overlap, so various fine registration algorithms for partially overlap point clouds emerge.

Chetveri-kov [6] proposed an ICP variant algorithm-Trimmed ICP(TrICP) for partial overlap of point clouds. The overlap degree is used to determine the proportion of point pairs with big distances to be removed, while point pairs with small distances are utilized to calculate the result. However, the overlap degree of point clouds needs to be specified manually. Li Xin [7] et al. extracted the overlap region by removing the point pairs with larger slope on the distance curve, but the slope threshold and other parameters needed to be determined by multiple tests. Liu Bin et al. [8] proposed to first extract the overlap region after encoding the point cloud using octree, and then calculate the transformation matrix using ICP method. Magnusson [9] proposes TheNormal Distributions Transform (NDT), which converts discrete point cloud data into continuous 3D probability density functions and uses Newton method to optimize point cloud registration. But its registration parameters need to be determined by many tests. Zaganidis et al. [10] made improvements to the NDT algorithm by exploiting the smoothness of the point cloud to partition it. By discarding a substantial number of points that were not divided into partitions, the algorithm focused on conducting NDT only on partitions of the same type instead of the entire point cloud.

Although the above methods can achieve fine registration of low overlap point clouds, the optimal parameters need to be provided manually or selected through experiments in the registration process, which is difficult to meet the automatic and fast registration of low overlap point clouds.

Hence, the present study introduces Auto-TrICP, an improved registration algorithm based on TrICP. The triangular threshold method is used to determine the distance threshold and bidirectionally merging of overlap regions, and then the validity factor is introduced to calculate the transformation matrix. The Auto-TrICP algorithm can accomplish rapid and automated registration of point clouds with precision.

2 Classic TrICP Algorithm Principle

The classical ICP algorithm can work out the result by finding the nearest neighbor points. However, in partially overlap point clouds, only wrong point pairs can be found when searching for matching points for non-overlap points [11], which undoubtedly increases the risk of mismatching.

To solve these problem, Chetverikov et al. proposed TrICP, which eliminated the excessively large point pairs by setting the overlap degree ξ, solved the transformation matrix with the remaining point pairs, and iterated several times until the stopping condition was met.

Given two 3D point sets, **P** and **U**, **P** have N_{P} points and p_i is a point in **P** for $i = 1, \ldots, N_{\mathrm{P}}$. Its steps are as follows:

1. For every point p_i in **P**, find its nearest neighbor m_i in **U**, and calculate the square of its distances d_i^2.

2. Sort all of the d_i^2, keeping the first N_{PO} point pairs and computing their sum as S_{TS}, where:

$$N_{\mathrm{PO}} = \xi N_{\mathrm{P}}. \tag{1}$$

3. Terminate the algorithm when the stop condition is met, otherwise continue to the next step.
4. Using the Singular Value Decomposition (SVD) solving the transformation matrix (R, t), and the objective function is follows:

$$(\mathrm{R}, \mathrm{t}) = \operatorname{argmin} \sum\nolimits_{i=1}^{N_{\mathrm{PO}}} \|m_i - (\mathrm{R}p_i + \mathrm{t})\| \tag{2}$$

5. Transform **P** according to follows and go to step 1:

$$p_i(\mathrm{R}, \mathrm{t}) = \mathrm{R}p_i + \mathrm{t} \qquad P(\mathrm{R}, \mathrm{t}) = \{p_i(\mathrm{R}, \mathrm{t})\}_1^{N_{\mathrm{P}}} \tag{3}$$

where, the algorithm is terminated when step 3 satisfies any of the following conditions:

- The iterations threshold N_{iter} has been reached.
- Point pair distance Mean Squared Error (MSE) value e is less than the given threshold, where:

$$e = S_{\mathrm{TS}}/N_{\mathrm{PO}}. \tag{4}$$

- The algorithm cannot further match the point cloud, that is, when the MSE value cannot be significantly reduced.

One limitation of TrICP is that the degree of overlap needs to be manually specified, which unavoidably adds complexity to the algorithm. Especially for more complex point clouds, the overlap degree is not easy to be determined, and it takes several tests to obtain more accurate overlap degree and registration results. Moreover, each iteration needs to calculate the distance curve once, which increases the time overhead. In addition, when using the traditional objective function for iteration, TrICP is still liable to be caught in local optimal solutions.

3 Improved TrICP Algorithm

Aiming at these problems of TrICP, we proposes an improved TrICP point cloud registration algorithm: Auto-TrICP, which uses the distance curve to determine the distance threshold, bidirectionally merges the overlap region, introduces the effectiveness factor, reduces the risk of getting trapped in local optimal solution, and realizes the fast and automatic registration.

3.1 Calculation of Distance Threshold

Firstly, the selection of distance threshold of overlap point clouds is discussed.

The distance threshold should separate overlap point clouds from non-overlap point clouds as far as possible. However, in distance curve, there is no strict dividing line between the two, but constitute the distance curve in the form of blending, in which the overlap part largely situate in the left side and the non-overlap part largely situate in the right side. In order to retain overlap point clouds as much as possible and provide sufficient point cloud information for subsequent registration, the distance threshold should not be too small, and in order to eliminate most non-overlap point clouds, the distance threshold should not be too large.

Common threshold selection algorithms include Otsu [12], iterative threshold method and triangular threshold method [13]. The three algorithms are all common methods in the field of image binarization, and can be used to binary classify curves. Otsu selects the number that makes the maximum inter-class variance between two categories as the threshold value, so that the difference between two categories is the greatest, but this algorithm is not suitable to deal with the situation where the proportion of two categories is very different. The iterative threshold method can select the threshold that minimizes the probability of error segmentation, but it requires the curve to have a more obvious valley. The trigonometric threshold method connects the highest point of the curve to the far point at both ends of the curve, and selects the point furthest away from the line on the curve below the line as the threshold value, as shown in Fig. 1(a). This method is suitable for the threshold calculation of unimodal function.

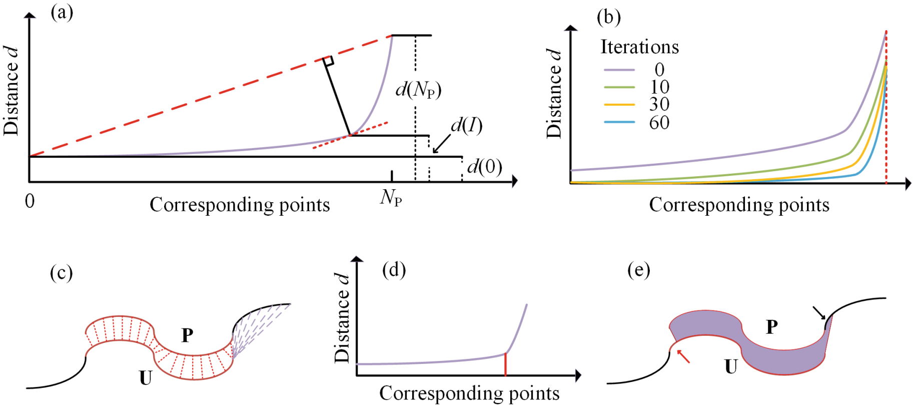


Fig. 1. (a) Distance curve and threshold selection diagram; (b) Distance curves of different iterations; (c) Sample point clouds; (d) Sample point clouds distance curve; (e) Clipping results of sample point clouds overlap areas

Observe the distance curve, the left point pair's distance and gradient is small, the right point pair's distance and gradient is large, which can be roughly fitted as an exponential function curve. In order to binary classify such functions, this paper proposes to use the triangular threshold method for segmentation, as shown in Fig. 1(a).

We hypothesise that distance curve is a smooth curve. The distance curve equation and the linear equation of the distance curve connected from end to end is:

$$d(i) = ae^{ki} + b, \tag{5}$$

$$[d(N_{\mathrm{P}}) - d(0)]i - N_{\mathrm{P}}d(i) + N_{\mathrm{P}}d(0) = 0. \tag{6}$$

The distance of the point pairs I with the maximum distance to the line can be calculated as:

$$d(I) = a\frac{e^{kN_{\mathrm{P}}} - 1}{kN_{\mathrm{P}}} + b \tag{7}$$

In order to verify whether $d(I)$ is reasonable as the distance threshold, figure out the ratio δ of the threshold distance to the maximum distance:

$$\delta = \frac{d(I) - d(0)}{d(N_{\mathrm{P}}) - d(0)} = \frac{e^{kN_{\mathrm{P}}} - 1 - kN_{\mathrm{P}}}{\left(e^{kN_{\mathrm{P}}} - 1\right)kN_{\mathrm{P}}} \tag{8}$$

In order to remove the point pairs with too large distance, δ should be small. δ is a monotonically decreasing function in regard to kN_{P}, and $\delta_{\max} = 0.5$. When $kN_{\mathrm{P}} = 3$, δ is about 0.281, and when $kN_{\mathrm{P}} = 4$, δ is about 0.231, we can find that this distance threshold can eliminate pairs of points that are too far apart.

After multiple iterations, as shown in Fig. 1(b), the overlap areas on the distance curve will gradually fit the horizontal coordinate, while the non-overlap areas will become steeper. The exponential function k value fitted by the distance curve will gradually increase, and the δ will gradually decrease, which means that the distance threshold can delineate the overlap areas more accurately.

For the two point clouds shown in Fig. 1(c), the triangular threshold method is used to determine the distance threshold which is exhibited in Fig. 1(d). Thus, the overlap region of the point clouds with low overlap is taken out which is exhibited in Fig. 1(e), and the purple region represents the overlap areas taken out, the red arrow represents the untaken part of the actual overlap areas, and the black arrow indicates the non-actual overlap areas contained in the extracted overlap areas. We can find that the overlap areas extracted in this way can contain most of the actual overlap regions.

3.2 Bidirectionally Merging of Overlap Point Clouds

So much for intercepting overlap areas in the TrICP algorithm, but only calculating the distance curve from point cloud **P** to **U** sometimes can not get a more ideal twopoint cloud overlap region. As shown in the middle of Fig. 2, no matter how to select distance threshold, the region pointed by the red arrow cannot be reached. However, when the distance curve from point cloud **U** to **P** is calculated, different results will be produced. Although it is still possible to obtain the region that does not belong to the actual overlap or there is an actual overlap region that is not taken, the real overlap region that cannot be selected from the overlap region calculated from point cloud **P** to **U** can be obtained.

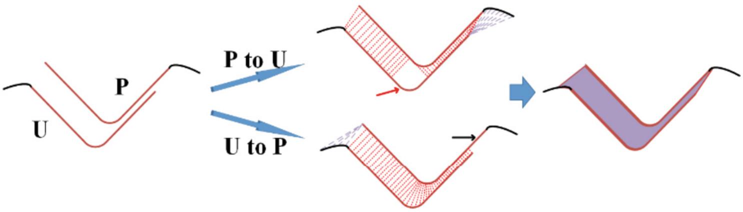


Fig. 2. Extraction process of overlap point cloud

Therefore, it is considered to merge the overlap areas obtained twice to obtain a more complete overlap areas, as shown in the purple region on the right side of Fig. 2.

The Fig. 3 shows the extraction process of overlap point clouds. In the overlap region obtained only by red point clouds to green point clouds, green point clouds will be severely absent, and the reverse is also similar. However, after combining the two points, the number of points will be approximately equal, and the two points will be close to the point clouds with real overlap.

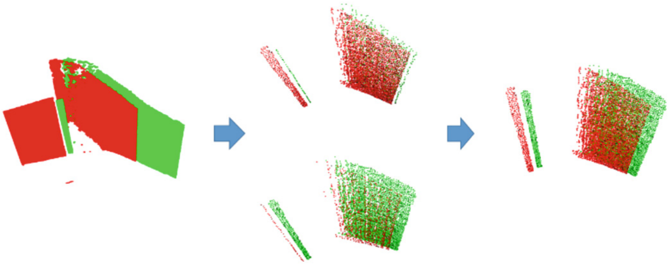


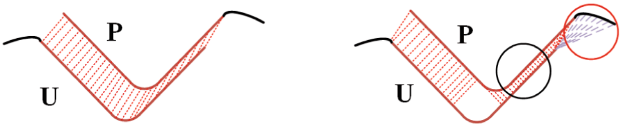
Fig. 3. Example of overlap areas extraction process

It is worth noting that getting the results of bidirectionally merging, duplicate points need to be removed to avoid affecting subsequent registration.

The advantage of obtaining such relatively complete overlap point clouds is that the same overlap point clouds could be used to calculate transformation matrices in subsequent iterations many times without affecting the registration effect. In the original algorithm, the overlap point clouds must be intercepted before each calculation of the transformation matrix, thus saving the algorithm running time.

3.3 The Introduction of Effectiveness Factor

After getting the overlap areas, the transformation matrix can be solved. At the early stage of iteration, many of the point pairs have not real correspondence. The experiment found that the point pairs in the middle of the distance curve have a higher contribution for outcome to converge to a correct value, but the point pairs with small distance, as shown in the circle of black in Fig. 4 (b), are more likely to lead to the convergence outcome being trapped in local optimal solutions. Moreover, the risk of point clouds in non-overlap areas increases sharply if their point pair distance is too large, as shown in the circle of red in Fig. 4 (b), which is an important factor in the result misregistration.



(a) Correct point pairs (b) Actual point pairs

Fig. 4. Comparison between the correct corresponding point pairs and the actual point pairs

In order to reduce the risk of falling into local optimality in registration and reduce the influence of the introduced non-overlap points, the effectiveness factor $w_i(0, 1)$ is introduced into the SVD as the weight of the point pair. As shown in Eq. (9), for points with small distance between the point pair, a smaller w_i is preferred to reduce the risk of falling into local optimality in registration. For point pairs with large distance, the probability of them being non-overlap point clouds is greatly increased. To reduce its influence, a smaller w_i is recommended. When the distance is moderate, the w_i should be larger.

$$(\mathrm{R}, \mathrm{t}) = \operatorname{argmin}\frac{1}{N_\mathrm{P}}\sum\nolimits_{i=1}^{N_\mathrm{PO}} w_i\|m_i - (\mathrm{R}p_i + \mathrm{t})\| \tag{9}$$

3.4 Auto-TrICP Algorithm Steps

The main steps of Auto-TrICP are as follows:

1. For each point p_i in **P**, find its nearest neighbor m_i in **U**, and calculate its distance d_i to get the distance curve.
2. Use the triangular threshold method to determine the distance threshold, and the overlap areas is intercepted by the distance threshold.
3. In the same way as step 1 and step 2, select the overlap point clouds calculated from **U** to **P**, bidirectionally merge and remove the duplicate points, and obtain the overlap areas required for registration.
4. Calculate the effectiveness factor for each point pair in a manner inversely proportional to the distance from the center of the overlap area of the curve.
5. Use the ICP method to register overlap areas several iterations, and the following judgments were made:

- Under the same overlap point cloud, if the iterations threshold n_iter has been reached, go back to step 1 and re-obtain the overlap areas.
- Continued iteration is difficult to effectively improve the registration accuracy, go back to step 1 and re-obtain the overlap region.
- Stop the algorithm if the MSE is small enough.
- In the whole algorithm process, the iterations threshold N_iter has been reached, the algorithm ends.

The improvements made in Auto-TrICP mainly focus on the following aspects: using the triangular threshold method to automatically estimate the overlap region; overlap areas are the result of bidirectionally merging; the effectiveness factor is introduced into the SVD; for reducing the times of overlap areas extraction, the overlap areas extraction is executed after multiple iterations of registration.

4 Experimental Analysis

In order to test the performance of the improved algorithm, we use the Auto-TrICP, the ICP and the TrICP to register low overlap point clouds. For point cloud data, a common Bunny point cloud and a more complex Armadillo point cloud were selected, and point cloud pairs with different overlap degrees were obtained after different angle clipoff. Among them, the Bunny point cloud is cropped at an oblique direction of 0°, 35° and 90°, and the Armadillo point cloud is cropped at an oblique direction of 0°, 45° and 90°. In the four groups of diagrams shown in Fig. 5, the first picture of each group is the original position of the point clouds, and the two point clouds are distinguished by yellow and green colors.

Because the TrICP needs to manually specify the overlap degree, this experiment uses the overlap degree corresponding to the distance threshold calculated by Auto-TrICP as the overlap degree of TrICP algorithm. After calculation, the overlap degree of Bunny0-35 is about 92%, Bunny0-90 is about 53%, Arm0-45 is about 66%, and Arm0-90 is about 56%.

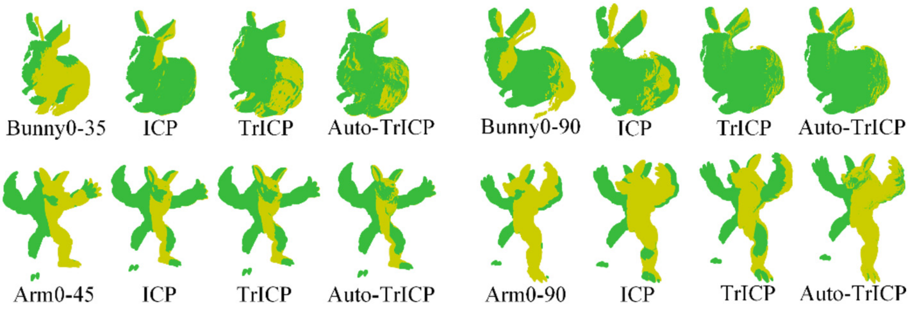


Fig. 5. Original point clouds and the results of each algorithm

Figure 5 shows the initial state of the point cloud and the registration results of each algorithm. The tighter the yellow-green point cloud blend, the better the registration effect. Intuitively, Auto-TrICP performs well for point clouds with different complexity and overlap. The ICP algorithm performs well on Bunny0-35 which has high overlap, but the registration effect on other point clouds with low overlap is very ordinary, and even gets wrong results. For TrICP algorithm, using the overlap degree provided by Auto-TrICP algorithm, it performs well on point clouds with high overlap such as Bunny0-35 and Arm0-45, and still performs well on point clouds with low overlap and high complexity such as Bunny0-90. However, on Arm0-90, a point cloud with low overlap and high complexity, the effect is inferior to that of the Auto-TrICP, which attests the effectiveness of the Auto-TrICP in reducing the risk of falling into the local optimal result. Therefore, Auto-TrICP has better results than the current algorithm in the complexity and overlap of different clouds.

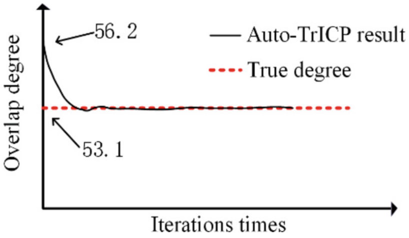


Fig. 6. The change curve of overlap degree of Auto-TrICP

In order to demonstrate the effectiveness of Auto-TrICP in extracting point cloud overlap degree, the calculated point cloud overlap degree values of Bunny0-90 with low overlap degree were recorded during Auto-TrICP operation and compared with the real value, as shown in Fig. 6. The results show that there is a certain difference between the calculated overlap degree and the real value in the initial operation of the algorithm, but the difference is small, and with the progress of iteration, the difference decreases rapidly and finally approaches the real value. At the beginning of the algorithm, two-point clouds were not registered, so it was difficult to obtain the exact overlap degree, but the small overlap degree difference would not affect the registration effect. With the progress of iteration, the overlap area of two-point clouds gradually matched, and the calculated overlap degree also approached the true value. Experiments show that this algorithm can obtain more accurate overlap degree, which is conducive to the subsequent registration.

Table 1. MSE value of each algorithm result

Piont cloud	ICP	TrICP	Auto-TrICP
Bunny0-35	0.00703	0.00562	0.00485
Bunny0-90	0.63251	0.00621	0.00615
Arm0-45	0.06843	0.00954	0.00856
Arm0-90	0.53642	0.06582	0.00942

In this paper, MSE value and algorithm execution time are selected as the quantitative evaluation indexes of each algorithm. Tables 1, 2 shows the MSE value of each algorithm result and the corresponding carry out time, in which the average value of the carry out time is taken from 5 experiments. As can be seen from Table 1, ICP algorithm does not consider eliminating non-overlap regions, resulting in a large MSE value. Although

Table 2. Execution time(s) of each algorithm result

Piont cloud	ICP	TrICP	Auto-TrICPAuto-TrICP
Bunny0-35	1.19	1.87	1.64
Bunny0-90	1.08	1.74	1.37
Arm0-45	1.56	2.31	2.03
Arm0-90	1.36	2.18	1.75

the algorithm takes the shortest time, it cannot make up for the lack of accuracy. TrICP algorithm has a small MSE value on Bunny0-35, Bunny0-90 and Armadillo0-45 point clouds, and has a good effect, while on Armadillo0-90 point clouds, the MSE value is large, and there is no good registration effect, and the algorithm execution time is the longest. Our algorithm is the smallest in all MSE values, and takes less time than TrICP. Especially on point clouds with low overlap, it will be more significantly optimized than TrICP algorithm in time, because point clouds with low overlap can eliminate more points.

5 Conclusion

In this paper, we propose an improved TrICP registration method: Auto-TrICP, which can automatically identify overlap regions of point clouds and perform more accurate registration. Aiming at the defect that the point cloud with low overlap degree needs to provide artificial overlap degree in the current algorithm, we propose a method to determine the distance threshold by using triangular threshold method, and form a complete overlap point clouds after bidirectionally merging. To solve the problem that current algorithms are easy to trap in local optimal solution, we propose a method to introduce the efficiency factor into the SVD, which can effectively reduce this risk. In view of the long carry out time of the current algorithms, the algorithm in this paper uses the relatively complete overlap areas multiple iterations to reduce the times of extracting overlap areas and improve the efficiency of the algorithm. Moreover, the efficiency improvement is more clear while the overlap degree is lower. In the future, we need to explore how to maintain the stability of the improved algorithm in the case of poor initial position.

References

1. Zhao, B.C., Yang, J.F., Wen, D.S., Gao, W., Chang, L.Y., Xue, B.: Chang'e-2 Lunar orbier CCD stereo camera design and validation. Spacecraft Eng. **20**(1), 14–21 (2011). (in Chinese)
2. Qiao, Y.J., Jia, B.M., Jiang, J.G., Wang, J.Y.: Networking method of multi-view stereo-vision measurement network. Infrared Laser Eng. **49**(7), 133–140 (2020). (in Chinese)
3. Liu, P., Golde, J., Morgenstern, J., et al.: Non-rigid point cloud registration for middle ear diagnostics with endoscopic optical coherence tomography. Int. J. Comput. Assist. Radiol. Surg. (2023)
4. Besl, P.J., McKay, H.D.: A method for registration of 3-D shapes. IEEE Trans. Pattern Anal. Mach. Intell. **14**(2), 239–256 (1992)

5. Wang, B., Liu, L., Hou, Y.Q., He, X.W.: Three-dimensional cardiac point cloud registration by improved iterative closest point method. Opt. Precis. Eng. **28**(2), 474–484 (2020). (in Chinese)
6. Chetverikov, D., Stepanov, D., Krsek, P.: Robust Euclidean alignment of 3D point sets: the trimmed iterative closest point algorithm. Image Vis. Comput. **23**(3), 299–309 (2004)
7. Li, X., Moster, Huang, H., Yang, S.J.: Multi-source point cloud registration method based on automatically calculating overlap. Infrared Laser Eng. **50**(12), 270–278 (2021) (in Chinese)
8. Liu, B., Guo, J.M., Deng, X.X.: A point cloud registration method based on Octree and ICP. Sci. Surveying Mapp. **41**(2), 130–132 (2016). (in Chinese)
9. Magnusson, M., Lilienthal, A., Duckett, T.: Scan registration for autonomous mining vehicles using 3D-NDT. J. Field Rob. **24**(10), 803–827 (2010)
10. Zaganidis, A., Magnusson, M., Duckett, T., Cielniak, G.: Semantic-assisted 3D normal distributions transform for scan registration in environments with limited structure. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4064–4069. IEEE, Canada (2017)
11. Rusinkiewicz, S., Levoy, M.: Efficient variants of the ICP algorithm. In: Proceedings Third International Conference on 3-D Digital Imaging and Modeling, pp. 145–152. Journal of the Japanese Society for Artificial Intelligence, Canada (2001)
12. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Trans. Syst. **9**(1), 62–66 (1979)
13. Zack, W.G., Rogers, E.W., Latt, A.S.: Automatic measurement of sister chromatid exchange frequency. J. Histochem. Cytochem.: Official J. Histochem. Soc. **25**(7), 741–753 (1977)